Secure Cloud Computing Framework with Automatic Intrusion Detection and Prevention in Cloud

M.Devi [1], Mrs.R.Uma, [2], Ms.M.Divya. [3]
Department of computer science, T.J.S Engineering College

Abstract- In this project we majorly offering real-time data security in cloud computing. Providing security to pet bytes of data is very important. A recent survey on cloud security

States that the security of users' data has the highest priority as well as concern. Therefore, to provide secure cloud framework I have proposed a system named secure cloud computing framework (SCCF). SCCF multi-layered security can protect data in real-time and it has three layers of security:

(i) Identity management and intrusion prevention
(ii) Secure encryption
(iii) Cloud storage

We propose a key management technique in cloud server. So, the proposed scheme greatly reduces the user's computation and storage overhead and makes full use of cloud server to achieve an efficient key management for the cryptographic cloud storage applications. Moreover, we introduce a key seed mechanism to generate a effective strength to the cloud data security. Our security analysis and performance evaluations both show that the proposed scheme is a secure and efficient key management protocol for the cloud storage applications with low overheads of computation and communication. In our project, the data owner file is encrypted using attribute based encryption technique and uploaded in a public cloud storage named Drop box. Also for data security in cloud storage, we have proposed Attribute based Encryption (ABE) algorithm during cloud storage. For cloud storage we have used Drop box. Also to enhance network security we have proposed automatic intrusion detection and prevention technique for 3 attacks namely Brute Force attack, SQL injection and Wrapping attack.

***Index Terms*—Cloud computing adoption framework (CCAF), security framework, Business Process Modeling Notation (BPMN), data security in the data center, multi-layered security protection**

1. INTRODUCTION

Cloud computing and its adoption has been a topic of discussion in the past few years. It Has been an agenda for organizational adoption due to benefits in cost-savings, improvement in work efficiencies, business agility and quality of services. With the

Rapid rise in cloud computing, software as a service (Saabs) is particularly in demand, since it offers services that suit users' need. For example, Health informatics can help medical researchers diagnose challenging diseases and cancers. Financial analytics can ensure accurate and fast simulations to be available for investors. Education as a service improves the quality of education and delivery. Mobile applications allow users to play online games and easy-to-use applications to interact with their peers. While more people and organizations use the cloud services, security and privacy become important to ensure that all the data they use and share are well protected. Some researchers assert that security should be implemented before the use of any cloud services in place. This makes a challenging adoption scenario for organizations since security should be enforced and implemented in parallel with any services. Although organizations that adopt cloud computing acknowledge benefits offered by cloud services, challenges such as security and privacy remain a scrutiny for organizational adoption. While overseeing the importance of security, the software engineering and development process should always design, implement and test security features.

The data centers have encountered challenges of rapid increase in the data. For example, in a data center that the lead author used to work with, daily increase of 100 terabytes of data was common. If the organization has encountered a rapid rise of data growth and is unable to respond quickly And efficiently, problems such as data traffic, data security and service level agreement issues can happen. In this paper, we focus on the data security while experiencing a large increase of data, whether they are from the external sources such as attack of viruses or Trojans; or they from the internal sources if users or clients accumulate hundreds of terabytes of data per day. This is a research challenge for data security which is essential for the better management of the data center to handle a rapid increase in the data.

Apart from the data center security management for rapid growth in data, the software engineering process should be robust enough to withstand attacks and unauthorized access. The entire process can be further consolidated with the development of a framework to tighten up the technical design and implementations, governance and policies associated with good practices. This motivates us to develop a framework, Cloud Computing Adoption Framework (CCAF), to help organizations successfully adopt and deliver any cloud services and projects.

LITERATURE REVIEW

Cloud Computing – Issues, Research and Implementations
Malden A. Vouch

Cloud‖ computing – a relatively recent term, builds on decades of research in virtualization, distributed computing, utility computing, and more recently networking,

Web and software services. It implies a service oriented architecture, reduced information technology overhead for the end-user, great flexibility, reduced total cost of ownership, on-demand services and many other things. This paper discusses the concept of ―cloud‖ computing, issues it tries to address, related research topics, and a
―cloud‖ implementation available today.

Security and Privacy Requirements Analysis within a Social Setting
Lin Liu, Eric Yu, John Mylopoulos

Security issues for software systems ultimately concern relationships among social actors - stakeholders, system users, and potential attackers - and the software acting on their behalf. This paper proposes a methodological framework for dealing with security and privacy requirements based on i*, an agent-oriented requirements modeling language. The framework supports a set of analysis techniques. In particular, attacker analysis helps identify potential system abusers and their malicious intents. Dependency vulnerability analysis helps detect vulnerabilities in terms of organizational relationships among stakeholders. Countermeasure analysis supports the dynamic decision-making process of defensive system players in addressing vulnerabilities and threats. Finally, access control analysis bridges the gap between security requirement models and security implementation models. The framework is illustrated with an example involving security and privacy concerns in the design of agent-based health information Systems. In addition, we discuss model evaluation techniques, including qualitative goal model analysis and property verification techniques based on model checking.

Review of "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud computing"
Jyothi, K., Ngai Reddy, B. Ravi Prasad

Cloud computing is an emerging computing paradigm in which resources of the computing infrastructure are provided as services over the Internet. As promising as it is, this paradigm also brings forth many new challenges for data security and access control when users outsource sensitive data for sharing on cloud servers, which are not within the same trusted domain as data owners. To keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when fine-grained data access control is desired, and thus do not scale well. The problem of simultaneously achieving fine-graininess, scalability, and data confidentiality of access control actually still remains unresolved. This paper addresses this challenging open issue by, on one hand, defining and enforcing access policies based on data attributes, and, on the other hand, allowing the data owner to delegate most of the

Computation tasks involved in fine-grained data access control to untrusted cloud servers without disclosing the underlying data contents. We achieve this goal by exploiting and uniquely combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption. Our proposed scheme also has salient properties of user access privilege confidentiality and user secret key accountability. Extensive analysis shows that our proposed scheme is highly efficient and provably secure under existing security models.

An Analysis of the Cloud Computing Security Problem

Mohamed Al Moray, John Grundy and Ingo Müller Cloud computing is a new computational paradigm that offers an innovative business model for organizations to adopt IT without upfront investment. Despite the potential gains achieved from the cloud computing, the model security is still questionable which impacts the cloud model adoption. The security problem becomes more complicated under the cloud model as new dimensions have entered into the problem scope related to the model architecture, multi-tenancy, elasticity, and layers dependency stack. In this paper we introduce a detailed analysis of the cloud security problem. We investigated the problem from the cloud architecture perspective, the cloud offered characteristics perspective, the cloud stakeholders' Perspective, and the cloud service delivery models perspective. Based on this analysis we derive a detailed specification of the cloud security problem and key features that should be covered by any proposed security solution.

Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data

Viper Goyal, Omkant Pandey, Amit Sanai, Brent Waters As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for NE-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE)

Assessing and Comparing Vulnerability Detection Tools for Web Services: Benchmarking Approach and Examples

Nano Attunes and Marco Vieira

Selecting a vulnerability detection tool is a key problem that is frequently faced by developers of security-critical web services. Research and practice shows that state-of-the-art tools present low effectiveness both in terms of vulnerability coverage and false positive rates. The main problem is that such tools are typically limited in the detection approaches implemented, and are designed for being applied in very concrete scenarios. Thus, using the wrong tool may lead to the deployment of services with undetected vulnerabilities. This paper proposes a benchmarking approach to assess and compare the effectiveness of vulnerability detection tools in web services environments. This approach was used to define two concrete benchmarks for SQL Injection vulnerability detection tools. The first is based on a predefined set of web services, and the second allows the benchmark user to specify the workload that best portrays the specific characteristics of his environment. The two benchmarks are used to assess and compare several widely used tools, including four penetration testers, three static code analyzers, and one anomaly detector. Results show that the benchmarks accurately portray the effectiveness of vulnerability detection tools (in a relative manner) and suggest that the proposed benchmarking approach can be applied in the field.

THEORETICAL BACKGROUND

In this project we majorly offering real-time data security in cloud computing.

Providing security to petabytes of data is very important. A recent survey on cloud security States that the security of users' data has the highest priority as well as concern. Therefore, to provide secure cloud framework I have proposed a system named secure cloud computing framework (SCCF).

SCCF multi-layered security can protect data in real-time and it has three layers of security:

(i) Identity management and intrusion prevention
(ii) Secure encryption
(iii) Multiple cloud storage

We propose a key management technique in cloud server. So, the proposed scheme greatly reduces the user's computation and storage overhead and makes full use of cloud server to achieve an efficient key management for the cryptographic cloud storage applications. Moreover, we introduce a key seed mechanism to generate an effective strength to the cloud data security. Our security analysis and performance evaluations both show that the proposed scheme is a secure and efficient key management protocol for the cloud storage applications with low overheads of computation and communication. In our project, the data owner file is split into 4 parts during file storage. 2 parts are encrypted using ABE algorithm and stored in Cloud server A. Other 2 parts are encrypted using ABE algorithm and stored in Cloud server B.

Also for data security in cloud storage, we have proposed Attribute based Encryption (ABE) algorithm during cloud storage. For cloud storage we have used Cloud Me. Also to enhance network security we have proposed automatic intrusion detection and prevention technique for 3 attacks namely Brute Force attack, SQL injection and Wrapping attack.

Proposed an agent-oriented modeling framework for analyzing security requirements. However, it is perceived as yet another modeling language than security requirements capturing framework. Provides a detailed definition and description on various cloud security data. Our proposed system involves automatic intrusion detection and prevention of security against Brute force, wrapping and SQL injection attacks. We proposed the multi-layered security to integrate security techniques to illustrate the essence and effectiveness of the framework. We have proposed our own framework, secure cloud computing framework (SCCF), to address the security challenge. The SCCF is a comprehensive model for adopting and applying cloud security principles systematically. ABE algorithm is proposed for data security in cloud. We introduce a key seed mechanism to generate an effective strength to the cloud data security. In our project, the data owner file is split into 4 parts during file storage. 2 parts are encrypted using ABE algorithm and stored in Cloud server A. Other 2 parts are encrypted using ABE algorithm and stored in Cloud server B.

METHODOLOGY

1. Identity Management
2. Intrusion Detection and prevention
3. Encryption
4. Key seed mechanism
5. Split and Merge
6. Multiple Cloud Storage

IDENTITY MANAGEMENT

- The identity management is divided into two roles: users and the security manager as follows.

- Users: Users can encrypt each key from his block and his own key. They can split files into blocks, encrypt them with the key, followed by signing the resulting encrypted blocks and creating the storage request. For each file, this key will be used to decrypt and rebuild the original file during the retrieval phase. The user also uses single sign-on to access each block with a compact signature scheme.

- Security Manager: Three roles are offered by the security manager. First, it can authenticate users during the storage/retrieval phase. Second, it can access control. Third, it can encrypt/decrypt data between users and their cloud.

INTRUSION DETECTION AND PREVENTION

- In this module automatic Intrusion detection system (IDS), encryption, deep packet inspection (DPI) and report the results to the controller. The main goal of Open Sec is to allow network operators to describe security policies for specific flows. The policies include a description of the flow, a list of security services that apply to the flow and how to react in case malicious content is found. The reaction can be to alert only, or to quarantine traffic or even block all packets from a specific source. Hence we have considered automatic intrusion detection and alerting network operator automatically when intruder tries brute force, SQL injection and wrapping attack.

Brute Force:

A password and cryptography attack that does not attempt to decrypt any information, but continue to try a list of different passwords, words, or letters. For example, a simple **brute-force attack** may have a dictionary of all words or commonly used passwords and cycle through those words until it gains access to the account. A more complex brute-force attack involves trying every key combination until the correct password is found. Due to the number of possible combinations of letters, numbers, and symbols, a brute force attack can take a long time to complete. The higher the type Of encryption used (64-bit, 128-bit or 256-bit encryption), the longer it can take.

SQL Injection:

SQL Injection is one of the most widely exploited web application vulnerability of the web era. SQL Injection is used by hackers to steal data from online businesses' and organizations' websites. This web application vulnerability is typically found in web applications which do not validate the user's input. As a result, a malicious user can inject SQL statements through the website and into the database to have them executed.

If a web application is vulnerable to SQL injection, a hacker is able to execute any malicious SQL query or command through the web application. This means he or she can retrieve all the data stored in the database such as customer information, credit card details, social security numbers and credential to access private areas of the portal, such as the administrator portal. By exploiting an SQL injection it is also possible to drop (delete) tables from the database. Therefore with an SQL Injection the malicious user has full access to the database.

Wrapping attack:

The attack uses a method known as XML signature wrapping and shows vulnerabilities while executing the web service request. In wrapping attack, the attacker tries to insert the malicious element in the SOAP (Simple Object Access Protocol) message structure in Transport

Layer Service (TLS) and after inserting the malicious code, fake content of the message is copied into the server and while executing, cloud server working is interrupted by the attacker.

ENCRYPTION:

- In our system we proposed cipher text-policy attribute-based encryption (CP-ABE) to address this problem, and give the first construction of such a scheme. In our system, a user's private key will be associated with an arbitrary number of attributes expressed as strings. On the other hand, when a party encrypts a message in our system, they specify an associated access structure over attributes. A user will only be able to decrypt a cipher-text if that user's attributes pass through the cipher-text's access structure.

KEY SEED MECHANISM

- Time-based Group Key Management algorithm for cryptographic cloud storage applications, which uses the proxy re-encryption algorithm to transfer major computing task of the group key management to the cloud server. So, the proposed TGKM scheme greatly reduces the user's computation and storage overhead and makes full use of cloud server to achieve an efficient group key management for the cryptographic cloud storage applications. Moreover, we introduce a key seed mechanism to generate a time-based dynamic Group key which effectively strengthens the cloud data security. Our security analysis and performance evaluations both show that the proposed TGKM scheme is a secure and efficient group key management protocol for the cloud storage applications with low overheads of computation and communication.

SPLIT AND MERGE:

Cloud Storage usually contains business-critical data and processes, hence high security is the only solution to retain strong trust relationship between the cloud users and cloud service providers. Thus to overcome the security threats, this paper proposes multiple cloud storage. Thus the common forms of data storage such as files and databases of a specific user is split and stored in the various cloud storages (e.g. Cloud A and Cloud B).

Databases consists of tables, rows and columns. Databases are easy to store in multiple cloud storages. Our application will act as a combiner and store different parts of the table such as rows and columns in multiple clouds using Vertical fragmentation and horizontal fragmentation. These rows and columns will be encrypted using RC4 (Stream Cipher) encryption algorithm. During response our application combines the data and sends to the verifier.

Files are stored in multiple clouds using cryptographic data splitting. The file is split into fragments and stored in distinct cloud servers with encrypted key. Thus once the authorized token for the specific file is requested, searchable encryption allows keyword search on encrypted data and combines the fragments. This is sent to the verifier.

CLOUD STORAGE:

Public cloud storage is a cloud storage model that enables individuals and organizations alike to store, edit and manage data. This type of storage exists on a remote cloud server and is accessible over the Internet. In this project, we proposed cloud me.

Public cloud storage is provided by a storage service provider that hosts, manages and sources the storage infrastructure publicly to many different users.

Public cloud storage service is also known as storage as a service, utility storage and online storage.

SOFTWARE ENVIRONMENT:

JAVA:

Initially the language was called as ―oak‖ but it was renamed as ―java‖ in 1995.The primary motivation of this language was the need for a platform-independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer's language
- Java is cohesive and consistent
- Except for those constraint imposed by the Internet environment. Java gives the programmer, full control

Finally Java is to Internet Programming where c was to System Programming.

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. In the areas of Security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Applications and applets. An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++ .Java's ability to create Applets makes it important. An Applet I san application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet I actually a tiny Java program, dynamically downloaded

Across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be react to the user input and dynamically change.

Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compilation of code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines
.This machine is called Java Virtual Machine.

Compiling and interpreting java source code. During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.

JVM:

The Java Virtual Machine is the cornerstone of the Java platform. It is the component of the technology responsible for its hardware- and operating system independence, the small size of its compiled code, and its ability to protect users from malicious programs. The Java Virtual Machine is an abstract computing machine. Like a real computing machine, it has an instruction set and manipulates various memory areas at run time. It is reasonably common to implement a programming language using a virtual machine; the best-known virtual machine may be the P-Code machine of UCSD Pascal. The first prototype implementation of the Java Virtual Machine, done at Sun Microsystems, Inc., emulated the Java Virtual Machine instruction set in software Hosted by a handheld device that resembled a contemporary Personal Digital Assistant (PDA). Oracle's current implementations emulate the Java Virtual Machine on mobile, desktop and server devices, but the Java Virtual Machine does not assume any particular implementation technology, host hardware, or host operating system. It is not inherently interpreted, but can just as well be implemented by compiling its instruction set to that of a silicon CPU. It may also be implemented in microcode or directly in silicon. The Java Virtual Machine knows nothing of the Java programming language, only of a particular binary format, the class file format. A class

File contains Java Virtual Machine instructions (or *byte codes*) and a symbol table, as well as other Ancillary information. For the sake of security, the Java Virtual Machine imposes strong syntactic and structural constraints on the code in a class file. However, any language with Functionality that can be expressed in terms of a valid class file can be hosted by the Java Virtual Machine. Attracted by a generally available, machine-independent platform, implementers of other languages can turn to the Java Virtual Machine as a delivery vehicle for their languages. The Java Virtual Machine specified here is compatible with the Java SE 7 platform, and supports the Java programming language specified in *The Java Language Specification, Java SE 7 Edition*.

Simple:

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while Simple types, such as integers, are kept as high-performance non-objects.

Robust

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

JSP:

The most significant of the many good reasons for this is that it is amazingly easy to develop sophisticated Web sites with JSPs. Anyone who can write HTML can quickly create rich, dynamic, and responsive Web sites that enable users to get the most out of their online time. Through a mechanism called JavaBeans, JSPs have made it possible for large teams or individuals working on complex projects to divide the work in such a way as to make each piece simple and manageable, without sacrificing any power. JSPs also provide a great deal of flexibility when generating HTML, through the ability to create HTML-like custom tags.

In addition to this fundamental ease of development, high-quality JSP tools are readily available and easy to use. Developers do not need to buy expensive software or commit to a particular operating system in order to use JSPs. The CD-ROM accompanying this book contains everything a JSP author needs to get started, and the tools are powerful enough to serve even a midsized Web site without problems. These free, open-source tools are stable and secure and run on nearly every platform. Of course, high-quality commercial JSP tools are available as well, suitable for serving even the most complex and high-traffic Web sites. Although JSPs have been useful and powerful since the beginning, this is an especially exciting time to be a JSP developer. The recently released version 2.0 of the JSP

Specification provides even more features that simplify the process of creating Web sites. In addition, a standard tag library that provides many JSP tags that solve a wide range of common problems has been released. Finally, in the time since they were released, a number of best practices for using JSPs have emerged. This book covers all the topics: the basic powerful features of the JSP specification, the improvements introduced with version 2.0, as well as the new standard tag library and all the things it does. In addition, this book discusses how best to use these tools, based on real-world experiences. However, before we get into all the fun, let's take a look back at how the Web has evolved. This will highlight the kinds of problems that Web authors have faced since the 12 beginning. Once this is Understood, it will be clear how JSPs solve these problems and make page creation so easy.

CSS:

While (X) HTML is used to describe the content in a web page, it is Cascading Style Sheets (CSS) that describe how you want that content to *look*. In the web design biz, the way the page looks is known as its presentation. CSS is now the official and standard mechanism for formatting text and page layouts. CSS also provides methods for controlling how documents will be presented in media other than the traditional browser on a screen, such as in print and on handheld devices. It also has rules for specifying the non-visual presentation of documents, such as how they will sound when read by a screen reader. Style sheets are also a great tool for automating production, because you can make changes to all the pages in your site by editing a single style sheet document. Style sheets are supported to some degree by all modern browsers.

Although it is possible to publish web pages using (X) HTML alone, you'll probably want to take on style sheets so you're not stuck with the browser's default styles. If you're looking into designing web sites professionally, proficiency at style sheets is mandatory.

SERVLETS:

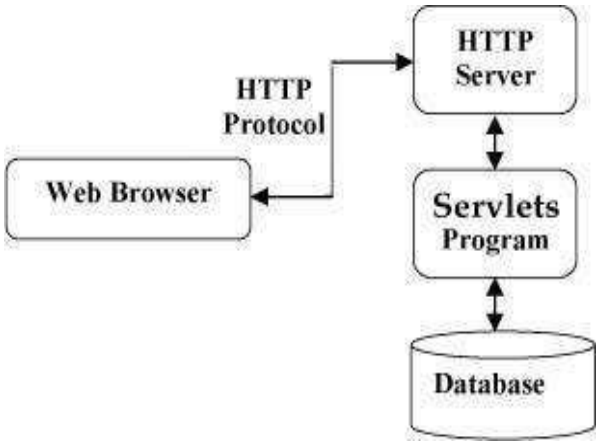
Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming

From a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI. Performance is significantly better. Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request. Servlets are platform-independent because they are written in Java. Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted. The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

SERVLET ARCHITECTURE:



Servlets perform the following major tasks:

- Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
- Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
- Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
- Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.
- Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (e.g., HTML),

Setting cookies and caching parameters, and other such tasks.

XML:

XML stands For **E**xtensible **M**arkup **L**anguage. It is a Text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and solutions:

- **XML is extensible:** XML allows you to create your own self-descriptive tags, or language, that suits your application.
- **XML carries the data, does not present it:** XML allows you to store the data irrespective of how it will be presented.
- **XML is a public standard:** XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

A short list of XML usage says it all:

- XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- XML can be used to exchange the information between organizations and systems.
- XML can be used for offloading and reloading of databases.
- XML can be used to store and arrange the data, which can customize your data handling needs.
- XML can easily be merged with style sheets to create almost any desired output.
- Virtually, any type of data can be expressed as an XML document.

IDE:

Programmers for software development. An IDE normally consists of a source codeDefined. Sometimes a version control system and various tools are integrated to simplify the construction of a Graphical User Interface (GUI). Many modern IDEs also have a class browser, an object browser, and class diagram, for use in object-oriented software development.

APACHE TOMCAT SERVER:

Tomcat is a Java servlet container and web server from the Jakarta project of the Apache Software Foundation (http://jakarta.apache.org). A web server is, of course, the program that dishes out web pages in response to requests from a user sitting at a web browser. But web servers

aren't limited to serving up static HTML pages; they can also run programs in response to user requests and return the dynamic results to the user's browser. This Is an aspect of the web that Apache's?

Tomcat is very good at because Tomcat provides both Java servlet and Java Server Pages (JSP) technologies (in addition to traditional static pages and external CGI programming). The result is that Tomcat is a good choice for use as a web server for many applications. And it's a very good choice if you want a free, open source (http://opensource.org/) servlet and JSP engine. Tomcat can be used stand-alone, but it is often used ―behind‖ traditional web servers such as Apache http, with the traditional server serving static pages and Tomcat serving dynamic

Servlet and JSP requests. No matter what we call Tomcat, a Java servlet container or servlet and JSP engine, we mean Tomcat provides an environment in which servlets can run and JSP can be processed. Similarly, we can absolutely say a CGI-enabled Web server is a CGI program container or engine since the server can accommodate CGI programs and communicate with them according to CGI specification. Between Tomcat and the servlets and JSP code residing on it, there is also a standard regulating their interaction, servlet and JSP specification, which is in turn a part of Sun's J2EE (Java 2 Enterprise Edition). But what are servlets and JSP? Why do we need them? Let's take a look at them in the following subsections before we cover them in much more detail in the future.

MY SQL:

MySQL is (as of March 2014) the world's second most widely used open-source relational database management

System (RDBMS). It is named after co-founder MichaelWidenius's daughter, My. The SQL phrase stands for Structured Query Language.The MySQL development project have made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For proprietary use, several paid editions are available, and offer additional functionality. Applications which use

MySQL databases include: TYPO3, MODx, Joomla, Word

Press, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-

profile, large-scale websites, including Google (though not for searches), Face book, Twitter, Flickr, and YouTube.

SQL YOG:

SQLyog is a GUI tool for the RDBMS MySQL. It is developed by Webyog, Inc. based out of Bangalore, India and Santa Clara, California. SQLyog is being used by more than 30,000 customers

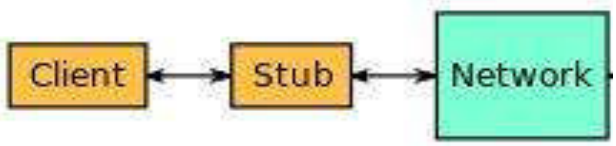
worldwide and has been downloaded more than 2,000,000 times.

RMI:

The **Java Remote Method Invocation** (**Java RMI**) is a Java API that performs the object-oriented equivalent of remote procedure calls (RPC), with support for direct transfer of serialized Java classes and distributed garbage collection.

1. The original implementation depends on Java Virtual Machine (JVM) class representation mechanisms and it thus only supports making calls from one JVM to another. The protocol underlying this Java-only implementation is known as Java Remote Method Protocol (JRMP).
2. In order to support code running in a non-JVM context, a CORBA version was later developed.

Usage of the term **RMI** may denote solely the programming interface or may signify both the API and JRMP, whereas the term RMI-IIOP (read:RMI over IIOP) denotes the RMI interface delegating most of the functionality to the supporting CORBA implementation.



SWING:

The AWT defines a basic set of controls, windows, and dialog boxes that support a Usable, but limited graphical interface. One reason for the limited nature of the AWT is that it translates its various visual components into their corresponding, platform-specific equivalents, or peers. This means that the look and feel of a component is defined by the platform, not by Java. Because the AWT components use native code resources, they are referred to as heavyweight. The use of native peers led to several problems. First, because of variations between operating systems, a component might look, or even act, differently on different platforms. This potential variability threatened the philosophy of Java: write once, run anywhere. Second, the look and feel of each component was fixed (because it is defined by the platform) and could not be (easily) changed. Third, the use of heavyweight components caused some restrictions like a heavyweight component is always rectangular and opaque. Swing was included as part of the Java Foundation Classes (JFC). Beginning with Java 1.2, Swing (and the rest of the JFC) was Swing does not replace AWT. Instead, Swing is built on the foundation of the AWT. This is why the AWT is still a crucial part of Java. Swing also uses the same event handling Mechanism as the AWT.

Two Key Swing Features:

Swing was created to address the limitations present in the AWT. It does this through two key features: lightweight components and a pluggable look and feel. Together they provide an elegant, yet easy-to-use solution to the problems of the AWT. More than anything else, it is these

Two features that define the essence of Swing.

DROP BOX:

Drop box is a cloud storage provider (sometimes referred to as an online backup service) that is frequently used as a file-sharing service. Drop box is a file hosting service operated by Drop box, Inc., headquartered in San Francisco, California, that offers cloud storage, file, personal cloud, and client software. Drop box allows users to create a special folder on their computers, which Drop box then synchronizes so that it appears to be the same folder (with the same contents) regardless of which computer is used to view it. Files placed in this folder are accessible via the folder, or through the Drop Box website and a mobile app. Drop box was founded in 2007 by Drew Houston and Arash Ferdowsi, as a Y Combinator startup company. Drop box provides client software for Microsoft Windows, Mac OS X, Linux, Android, iOS, BlackBerry OS and web browsers, as well as unofficial ports toSymbian, Windows Phone, and MeeGo.

Java Networking:

The term *network programming* refers to writing programs that execute across multiple devices (computers), in which the devices are all connected to each other using a network.

The java.net package of the J2SE APIs contains a collection of classes and interfaces that provide the low-level communication details, allowing you to write programs that focus on solving the problem at hand.The java.net package provides support for the two common network protocols:

TCP:

TCP stands for Transmission Control Protocol, which allows for reliable communication between two applications. TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.

UDP: UDP stands for User Datagram Protocol, a connection-less protocol that allows for packets of data to be transmitted between applications.This tutorial gives good understanding on the following two subjects:

Socket Programming: This is most widely used concept in Networking and it has been explained in very detail.

URL Processing: This would be covered separately. Click here to learn about URL Processing in Java language.

Socket Programming:

Sockets provide the communication mechanism between two computers using TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.

When the connection is made, the server creates a socket object on its end of the communication. The client and server can now communicate by writing to and reading

from the socket. The java.net.Socket class represents a socket, and the java.net.ServerSocket class provides a mechanism for the server program to listen for clients and establish connections with them.

The following steps occur when establishing a TCP connection between two computers using sockets:The server instantiates a ServerSocket object, denoting which port number communication is to occur on.The server invokes the accept() method of the ServerSocket class. This method waits until a client connects to the server on the given port.After the server is waiting, a client instantiates a Socket object, specifying the server name and port number to connect to.The constructor of the Socket class attempts to connect the client to the specified server and port number. If communication is established, the client now has a Socket object capable of communicating with the server. On the server side, the accept () method returns a reference to a new socket on the server that is connected to the client's socket. After the connections are established, communication can occur using I/O streams. Each socket has both an OutputStream and an InputStream. The client's OutputStream is connected to the server's InputStream, and the client's InputStream is connected to the server's OutputStream.TCP is a twoway communication protocol, so data can be sent across both streams at the same time. There are following usefull classes providing complete set of methods to implement sockets.

CONCLUSION:

Our project has demonstrated the CCAF multi-layered security for the data security in the Data Centre under the proposal and recommendation of CCAF guidelines. We explained the rationale, overview, components in the CCAF, where the design was based on the requirements and the implementation was illustrated by its multi-layered security. In this we have defined automatic intrusion detection and prevention for 3 attacks namely brute force, SQL injection and wrapping attack as predefined software policies. Finally the user data are encrypted and stored in the public cloud. Our approach provides an integrated solution to cloud security based on a clear framework, business process modelling to study the impact on the performance of a user while accessing the service.

REFERENCES:

[1] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi,―Cloud computing – The business perspective,‖ Decision Support Syst., vol. 51, no. 1, pp. 176–189, 2011.

[2] M. A. Vouk, ―Cloud computing—issues, research and implementations,‖ J. Comput. Inf. Technol.—CIT, vol. 4, pp. 235–246, 2008.

[3] A. K. Jha, C. M. DesRoches, E. G. Campbell, K. Donelan, S. R. Rao, T. G. Ferris, and D. Blumenthal, ―Use of electronic health records in US hospitals,‖

New England J. Med., vol. 360, no. 16, pp. 1628–1638, 2009.

[4] H. T. Peng, W. W. Hsu, C. H. Chen, F. Lai, and J. M. Ho, ―Financial cloud: open cloud framework of derivative pricing,‖ in Proc. Int. Conf. Social Comput., Sep. 2013, pp. 782–789.

[5] M. Mircea and A. I. Andreescu, ―Using cloud computing in higher education: A strategy to improve agility in the current financial crisis,‖ Commun. IBIMA, vol. 2011, pp. 1–15, 2011.

[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, ―Above the clouds: A Berkeley view of cloud computing,‖ Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.